Particle Based Image Segmentation with Simulated Annealing

Everts, M.H.; Bekker, H.; Jalba, A.C.; Roerdink, J.B.T.M.

Published in:
EPRINTS-BOOK-TITLE

Publication date:
2007

Particle Based Image Segmentation with Simulated Annealing

M.H. Everts H. Bekker A.C. Jalba J.B.T.M. Roerdink

Department of Mathematics and Computing Science,
University of Groningen,
P.O. Box 800, 9700 AV Groningen, The Netherlands
{maarten,bekker,andrei,roe}@cs.rug.nl

Keywords: Image segmentation, particle systems, simulated annealing

Abstract

The Charged Particle Model (CPM) by Jalba et al., is a physically motivated deformable model for shape recovery and segmentation. It simulates a system of charged particles moving in an electric field generated from the input image, whose positions in the equilibrium state are used for curve or surface reconstruction. As an alternative we propose to use simulated annealing to position these particles and present a fast and simple algorithm that is able to match CPM's results on images with well defined contours.

1 Introduction

An important part of the analysis of images and volume data is segmentation and shape analysis. Segmentation is the process of determining regions in an image (be it 2D or 3D) that have some particular property, for example intensity or texture. Segmentation is difficult, not only because input data often contains noise and sampling artifacts, but more so because it is hard to precisely formulate segmentation criteria.

This is why there is a wide selection of (often application specific) image segmentation techniques available. Very popular are the methods that can be categorized as deformable models where curves (or surfaces) move in the image domain under the influence of internal and external forces in search of a low energy configuration. The internal forces keep the curve smooth, the external forces move the curve towards features in the input image (e.g., edges).

Two types of deformable models can be distinguished [9]: (1) parametric deformable models and (2) geometric deformable models. In parametric deformable models, curves (and surfaces) are explicitly represented and manipulated during deformation. This makes them well suited for direct interaction. However, parametric deformable models are sensitive to initialization, have trouble moving into concavities, and most of them cannot change topology whilst deforming.

Geometric deformable models or level set methods on the other hand can adapt to changing topology easily. The central idea of these methods is that a shape (e.g., curve or surface) is embedded as the zero level set of a higher-dimensional scalar function. Through this function the shape is implicitly manipulated [8]. Many variations exist, including versions that can handle texture [2, 7] and prior (shape) information [3].

A recent variation of the deformable model methods is Particle Based Segmentation (PBS). Here segmentation is a two-stage process. In the first stage, particles are positioned in the image domain with sufficient density. In practice, like in the methods mentioned earlier, this means finding a low energy configuration in some sense. In the second stage one or more shapes are (re)constructed from the positions of the particles. Obviously, the quality of the first stage strongly influences the feasibility of the second stage.

The Charged Particle Model (CPM), introduced in [4] is an example of PBS. In CPM particles are positioned by means of a Molecular Dynamics (MD) like simulation of charged particles in a potential field generated from the input image. To be able to better cope with noise in the images and to improve efficiency, CPM uses a multi-scale approach with a Gaussian pyramid. In this paper we will try to simplify the process of positioning the particles by using simulated annealing instead of MD. The main question we will try to answer is whether it is possible to use simulated annealing to obtain similar results as the CPM method. Additionally, we explore different particle types and a simpler way of calculating a potential field. For the curve and surface reconstruction we use

the same existing methods as CPM.

2 Charged Particle Model

The Charged Particle Model (CPM) is an image segmentation method inspired by classical electrostatics. It simulates a system of positively charged particles moving in an electrostatic field generated by fixed negative charges placed at each pixel position in the input image, with charge magnitude proportional to the edgemap of the input image. The edgemap is defined as

$$f(x,y) = \left|\nabla(G_\sigma(x,y) * I(x,y))\right|, \qquad (1)$$

where G_σ is a Gaussian kernel and I the input image. This means the positively charged particles are attracted to the edges in the input data. The charged particles also repel each other and this ensures that the particles are distributed evenly on the contours once the system of moving particles reaches an equilibrium state.

The relative influence of the two forces - from the electrostatic field and from the inter-particle interaction - are controlled by a parameter. Knowing the forces on the particles, the Newtonian equations of motion are integrated over time. To be able to reach an equilibrium state a damping factor β is added.

Even though the CPM paper [4] focuses on two-dimensional images, it shows that CPM also works for three-dimensional input.

3 Simulated annealing

As shown above, in CPM the equilibrium state, a low energy state, is obtained by simulating a dynamic system; the Newtonian equations of motion with damping are integrated over time. So besides a position each particle also has a velocity, but for the reconstruction part only the positions are used. We can simplify this process by reformulating this part of the segmentation process as a combinatorial optimization problem, for which various solutions exist.

Consider a system of N identical particles where x_i denotes the position of particle i and where the system state X is defined as $X \equiv \{x_1, x_2, ..., x_N\}$. The particles move in a fixed potential field $\phi_f(x)$, and between every pair of particles i and j a potential $\phi_p(x_i, x_j)$ is given. So the total potential energy of the particle system is

$$\phi(X) \equiv \Sigma_i \phi_f(x_i) + \Sigma_i \Sigma_{j>i}\ \phi_p(x_i, x_j). \qquad (2)$$

For PBS it is required that the total potential energy is minimized, so in principle we are looking for

$$\{X_{\min} \mid \phi(X_{\min}) \leq \phi(X)\ \ \forall\ X\}.$$

It is however not computationally feasible to calculate $X_{\min}$ exactly, so we will be satisfied with finding a near minimum state X_{nm} that has approximately minimal energy.

1: $\hat{X}_{i+1} \leftarrow$ GetNewState(X_i)
2: $\Delta E \leftarrow E(\hat{X}_{i+1}) - E(X_i)$
3: **if** $\Delta E \leq 0$ **or** $p(\Delta E, T) >$ random$(0 \ldots 1)$ **then**
4: $\quad X_{i+1} \leftarrow \hat{X}_{i+1}$ {Accept}
5: **else**
6: $\quad X_{i+1} \leftarrow X_i$ {Reject}
7: **end if**

Figure 1: Pseudo-code for one iteration of the simulated annealing algorithm.

Simulated annealing (SA) is a method for finding an approximation of the global minimum for combinatorial optimization problems like the one stated above. This probabilistic algorithm has its roots in the Metropolis Monte Carlo technique.

Using SA for positioning particles instead of a dynamic approach like in CPM has a number potential advantages. First, by not needing velocity, the search space becomes smaller and this could improve the convergence speed. Also, SA's ability to look beyond local minima makes the segmentation method suitable for automatic segmentation as the whole image is sampled. Finally, and perhaps more importantly, using an SA approach results in various simplifications. Not only is the SA algorithm itself short and simple (as we will shown next), force evaluations are also no longer needed, only potentials.

At each step of the SA algorithm a random nearby state $\hat{X}_{i+1}$ is generated from the current system state X_i. A downhill move (a state with lower energy) is always accepted, an uphill move is accepted with a probability $p(\Delta E, T)$ that depends on the energy difference between the states and on a global parameter T, serving as a fictitious temperature that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is high, but increasingly goes downhill as T goes to zero. The allowance for uphill moves prevents the method from becoming stuck at local minima. Figure 1 shows the short and simple pseudo-code of the algorithm. For accepting or rejecting uphill trial moves we use the standard Metropolis criterion [6] given by

$$p(\Delta E, T) = e^{\frac{-\Delta E}{cT}}, \qquad (3)$$

where c is a constant.

The above is a general description of the SA algorithm. For our goal, segmentation by SA, two decisions have to be made: how to generate a new state and what annealing (cooling) schedule to use.

1: $T = 0$
2: Equilibrate the system for 10 temperature steps
3: $T = 0.01$
4: **while** uphillMoves/moves < 0.1 **do**
5: Do one temperature step and count the number of uphill moves
6: $T = 2 * T$
7: **end while**
8: $T_0 = T/2$ {Undo last temperature increase}

Figure 2: Pseudo-code for determining T_0.

3.1 Annealing schedule

The goal of SA is approximating the optimum of a high dimensional function. For this it is not required that during SA the system is quasi static, it is only required that, at every temperature, parameter space is sampled with sufficient density.

For discussing the role of temperature in SA we will use the following terminology. A trial move is an attempt to move a particle to a new position. A temperature step is a series of s trial moves at a constant temperature. We take $s = 10N$, where N is the number of particles.

The temperature at the start of the simulation (T_0) should be high enough, such that the system can step over local minima, i.e., such that a certain percentage of uphill moves are accepted.

We propose to use the system's behavior at different temperatures to determine T_0. First set the temperature to zero and simulate a few steps at this temperature to equilibrate the system. Then choose a low temperature T_l and run the simulation while doubling T_l until the number of uphill moves is 10% of the total number of moves. That temperature is used for T_0. Figure 2 shows the pseudo-code for this scheme.

The SA is started with T_0 and in M steps the temperature is lowered to T_f, the final temperature. For the cooling schedule we will use the widely accepted proportional cooling schedule [5],

$$T_i = T_{i-1}\alpha, \quad \alpha = \left(\frac{T_f}{T_0}\right)^{1/M}. \tag{4}$$

3.2 Generating a new state

For our particle system, a simple and effective procedure for generating a new trial system state is randomly choosing one of the particles and adding a random displacement to it. This displacement should be large when the temperature is high and small when the temperature is low. So we let the maximum displacement $d_{\max}$ depend on the temperature as

$$d_{\max} = c_1 * T, \tag{5}$$

where c_1 is a constant that represents the maximum displacement at the start of the simulation. To ensure the search space is sufficiently sampled, we suggest to choose c_1 such that at the start of the simulation the maximum displacement is larger than half of the largest image dimension.

The actual displacement is calculated by generating a random length ($\leq d_{\max}$) and a random direction.

3.3 Grid based approach

Positions of particles in CPM are real-valued vectors, i.e., particles can position themselves anywhere in the input data space. For SA a similar approach would be possible, but (initially) restricting the particle positions to integer values results in a smaller search space, and thus faster convergence.

In CPM, using integer positions would not make sense because in CPM the low energy state is found by carefully following trajectories. Rounding to integer positions would cancel all that effort.

Besides faster convergence, there are additional advantages. First of all, simple (fast) integer arithmetic can be used. For evaluation of the potential no interpolation is necessary. Second, even more importantly, a grid based approach simplifies the range query for neighbor finding. As positions are integer values, we can simply mark on a grid G where the particles are. Finding the nearest neighbors of a particle resolves to looking at grid points in G within the cut-off radius r_C. Neighbor finding in CPM is done by constructing a kd-tree each timestep and querying it for each particle. This accounts for a significant part of the running time.

Of course r_C should be kept small as increasing it increases the number of evaluated grid-points per step quadratically.

4 From input to potential

The electric field in CPM is obtained by a number of steps. First the edgemap of the input image is calculated using eq. (1). The edgemap is mapped to a grid of fixed charges for which a potential field is calculated. Finally the electric field is obtained by computing the gradient of the potential field. Figure 3 illustrates this process schematically.

For an SA approach only energies need to be evaluated. For that reason we could just use the edgemap as potential, skipping the step of assigning values to charges and calculating the potential field. This simplifies the pre-processing step greatly. Furthermore, calculating the potential from charges is an expensive operation, in particular when the method is scaled to three dimensions.

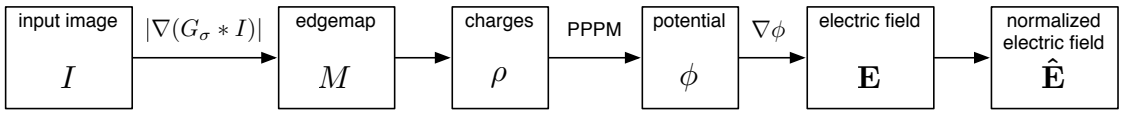


Figure 3: CPM overview

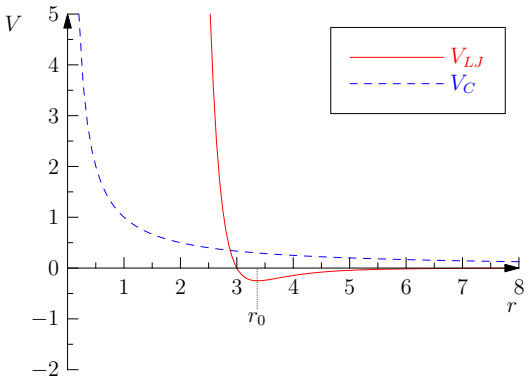


Figure 4: Graphs of Coulomb and Lennard-Jones potential.

5 Particle types

In section 3 on simulated annealing the inter-particle potential ϕ_p was not yet specified. We can turn to physics theory for suggestions. In CPM, the Coulomb potential from electrostatics is used, i.e., the potential between two particles is proportional to the inverse of the inter-particle distance r:

$$V_C = \frac{1}{4\pi\epsilon_0}\frac{q_1 q_2}{r}. \tag{6}$$

The Lennard-Jones (LJ) potential is used to model the long range attractive force and the short range repulsive force of atoms and molecules. It is defined as

$$V_{LJ} = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{7}$$

where σ is the distance for which the potential is zero. The $(\frac{\sigma}{r})^{12}$ term describes short range repulsion and the $(\frac{\sigma}{r})^{6}$ term describes the long range attraction. Figure 4 shows the graphs of the Coulomb potential and the LJ potential (for $\sigma = 3$) respectively. In the LJ graph one can see that the first derivative is zero for a value of r slightly larger than σ. This means that this r_0 is the distance particles try to maintain.

The difference in behavior is illustrated by figure 5. In an SA simulation where a Coulomb potential is used the particles are distributed evenly on the edge of the object, as shown in figure 5(a). In a similar simulation using LJ potential the particles tend to cluster together at approximately distance r_0. The interesting property of LJ potential is that one can directly influence the (minimum) distance between particles.

6 Implementation

6.1 Particle initialization

For now we will assume there are enough particles at the start of the simulation to segment the objects. This still needs to be studied in more detail. Options are pre-calculating the number of particles needed or a dynamic add and removal scheme like the one CPM uses.

When the LJ potential is used, the simulation must start with exactly the right number of particles, whereas when the Coulomb potential is used the particles will spread out over the edges. This is why we use the Coulomb potential for the experiments we describe in this paper.

At the start of the simulation the positions are just randomly distributed over the input image.

6.2 Refinement

A grid based implementation returns integer particle positions and as such the reconstructed curve or surface through these particles can show artifacts. To remedy this we will do a refinement step. We have experimented with two different refinement approaches.

The first is allowing half grid positions for small T, which basically resolves to doubling the size of the grid dimensions of G. The potential for particles not on grid-positions that do not resolve to image positions can be interpolated.

The other refinement technique is somewhat ad-hoc. At the end of the simulation we do the following. For each particle we calculate the center-of-mass of the neighboring pixels (including the one on the particle's position) in the potential map and then just move it there. The result is that if a particle is on a two-pixel wide edge it will be moved to the center of that edge.

This last method appears to give the best results (see figure 10) and is fast. So this is what we used to obtain the results showed in the next section.

6.3 Curve reconstruction

Curve reconstruction from unorganized points is a well studied problem and like CPM, we use the algorithms by Amenta et al. [1] for this.

7 Results

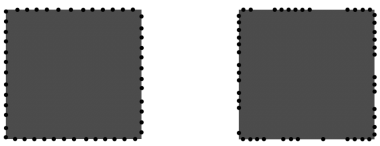
In this section we will show results of our SA CPM variation on a number of synthetic and non-synthetic grayscale images. For all these simulations we used the Coulomb potential for inter-particle interaction because, while the Lennard-Jones potential works, it requires that the simulation starts with exactly the right number of particles. The number of temperature steps (M in eq. (4)) for these experiments is 50.

We will start by answering the question whether we can reproduce the CPM's results using SA. Figures 6, 7, and 8 show simple synthetic input images combined with the reconstructed curves. Figure 7 for example is an image that is hard for parametric deformable models like snakes because of the concavities. The spiral of figure 8 has an even more extreme concavity and SA produces a proper smooth curve that captures the object in the image. With SA particles can hop right to the center of the spiral, whereas - if it were not for the pyramid approach - in CPM particles will have to follow the spiral to cover the contours.

So our results indicate that for synthetic images the results of SA are just as good as those of CPM. For non-synthetic (natural) images it is another story. Figures 11 and 12 show that while the results are similar, CPM's results appear to be better. In CPM's results there are more closed curves and less small curves resulting from noise.

To show that this is not the result of using a different potential field (see section 4), we have done SA experiments using both the proposed simplified potential, as well as the potential calculated from fixed charges (like CPM). Figure 9 shows that the difference between the two approaches for defining the potential is small.

One area in which SA does excel is speed. In the grid-based SA implementation particles quickly converge to a low energy configuration. Timings have shown that on average, our SA implementation is about twice as fast as CPM with a multi-scale approach.



(a) Coulomb (b) Lennard-Jones

Figure 5: The type of potential used for inter-particle interaction influences the behavior of the particles. When the Coulomb potential is used particles spread out evenly on the edges (a). Using the Lennard-Jones potential makes the particles cluster together with a certain distance (b).

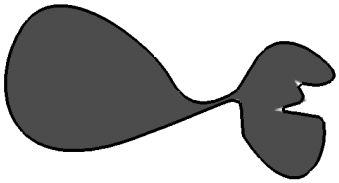


Figure 6: The curve constructed from particles positioned with SA together with the input image.

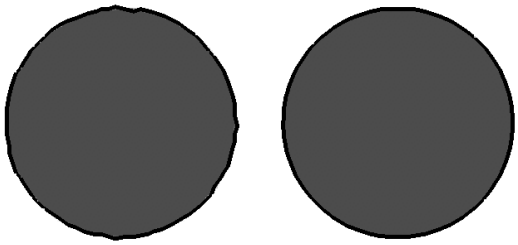
(a) CPM (b) SA

Figure 7: The input image, a simple image with concavities, shown together with the constructed curves created with CPM (a) and SA (b).

Figure 8: The curve constructed from particles positioned with SA shown together with the input image, a spiral with a deep concavity.

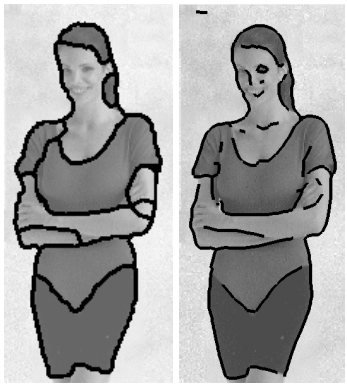
(a) CPM (b) SA with potential from charges (c) SA with edgemap as potential

Figure 9: The difference between using the original potential calculated from charged (b) and using the edgemap as potential (c) is small on this natural image.



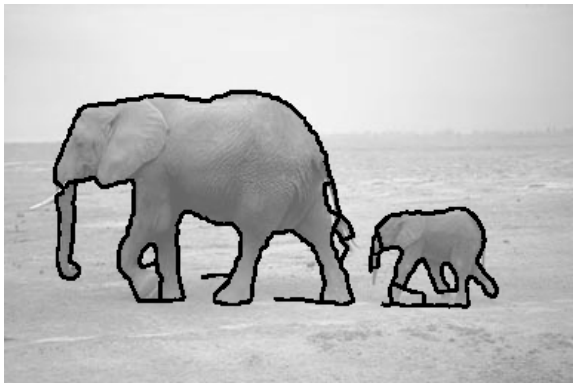
(a) Without refinement (b) With refinement

Figure 10: Refinement of the integer positions of the particles results in a smoother curve.

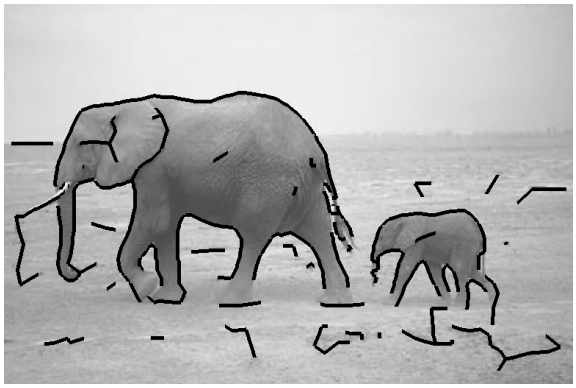


(a) CPM (b) SA

Figure 11: The reconstructed curves (shown together with the input image) from CPM and SA are similar, but not quite the same.



(a) CPM



(b) SA

Figure 12: Another natural image that shows the difference between the curves produces by CPM and SA: in SA's result there are more small noisy curves and the curves are not as closed as in CPM's result.

8 Conclusion

The main question we tried to answer in this paper was whether it is possible to reproduce CPM's results using simulated annealing. Reasons for considering SA included that SA would result in a considerable simplification and that convergence to a solution could be faster. We found that for images with well defined edges, SA works as well as CPM. As expected, we found that the SA implementation is faster than CPM; approximately twice as fast. For images that do not have well defined edges (like natural images) the results differ; for SA the results have more small noisy curves and the main curves are not as closed as those produced by CPM.

We also experimented with the Lennard-Jones potential as an alternative for the Coulomb potential for inter-particle interaction. We found that the LJ potential has the advantage that it allows us to directly influence the particle distances, but also that it has the disadvantage that a simulation has to start with the correct number of particles.

Finally, we showed that by using SA the potential calculation can be simplified by using the edgemap as the potential.

References

[1] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, New York, NY, USA, 1998. ACM Press.

[2] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, February 2001.

[3] A. Foulonneau, P. Charbonnier, and F. Heitz. Affine-invariant geometric shape priors for region-based active contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1352–1357, 2006.

[4] A. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink. CPM: A deformable model for shape recovery and segmentation based on charged particles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1320–1335, 2004.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[7] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vision*, 46(3):223–247, 2002.

[8] J. S. Suri, K. Liu, S. Singh, S. N. Laxminarayan, X. Zeng, and L. Reden. Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review. *IEEE Trans Inf Technol Biomed*, 6(1):8–28, Mar 2002.

[9] C. Xu, D. L. Pham, and J. L. Prince. *Medical Image Segmentation Using Deformable Models*, chapter 3, pages 129–174. SPIE Press, 2000.